Hosting virtual de alta disponibilidad con Apache

Enric Torrente Palacios

Resumen— El crecimiento de la demanda actual de servicios de alojamiento web, ha desencadenado en una gran cantidad de empresas que ofrecen múltiples servicios de hosting, con distintas características que influyen en la elección del servicio al cliente final. Hoy en día, factores como la disponibilidad del servicio, pueden ser importantes para la elección del sistema de alojamiento. Por ello, las empresas dedicadas a ofrecer este tipo de servicios, utilizan diversas tecnologías en sus sistemas de alojamiento que ayudan a mantener un cierto grado de disponibilidad. En el desarrollo de este proyecto, se muestra el proceso de implementación de un servicio de hosting compartido, mediante la utilización de servidores web basados en Apache y combinando tecnologías como HAProxy, KeepAlived y MySQL NDB Cluster para la formación de clústeres. Bajo este entorno también se realizan pruebas de rendimiento y se analiza la disponibilidad del sistema. Adicionalmente, se incluye un panel de control a medida, que facilita al usuario final la gestión del servicio de manera sencilla y transparente.

Palabras clave— Apache, alta disponibilidad, HAProxy, clúster MySQL, Keepalived, Bind DNS, monitorización Zabbix, servicio alojamiento web, clusterización, PHP, balanceo de carga, panel de control para hosting, código abierto.

Abstract—The growth in the demand of web hosting services, has lead into multiple companies offering a wide range of hosting services, based on different technologies that fulfill client requirements. Numerous factors, as the availability of the service, influence in the client final choice of the proper hosting. Due to this fact, companies providing these kind of services, apply the usage of mixed technologies into their systems willing to achieve concrete degrees in the availability of their services. This project covers the whole implementation process of a web shared hosting service, making usage of Apache web services and combining technologies such as HAProxy, KeepAlived and MySQL NDB Cluster, which help in accomplishing clustering formation. Performance tests and analytics proof the availability of the system. Additionally, a customized web control panel tool is provided, to ease the administration of the system by the end customer.

Index Terms— Apache, high availability, HAProxy, MySQL cluster, Keepalived, Bind DNS, Zabbix monitoring, web hosting service, clustering, PHP, load balancing, hosting control panel, open source.

1 Introducción

SON numerosas las empresas que hoy en día tienen presencia en Internet. Existen diferentes recursos para alcanzar este objetivo, como las redes sociales, los directorios de empresas o la creación de un sitio web bajo un dominio propio.

Cuando se opta por la creación de un sitio web, se deben tener en cuenta, entre otros aspectos, la disponibilidad, los usuarios simultáneos que se pueden abastecer o las tecnologías utilizadas, antes de contratar un servicio de alojamiento web o implementar nuestro propio servidor.

En el siguiente documento, se muestran los aspectos fundamentales de un servicio de alojamiento web, llevando a cabo la implementación de un sistema de hosting compartido. Para ello, se ha estructurado los contenidos del siguiente modo.

Primeramente, se muestra el contexto del proyecto con la situación actual y las tecnologías que se emplean en estos sistemas. Acto seguido, se definen los objetivos principales que se quieren alcanzar en este proyecto, junto con las tecnologías que se emplearán, detallando posteriormente, la planificación y metodología llevadas a cabo.

Para su implementación, en primer lugar, se presenta el diseño de red y las máquinas que formarán la infraestructura, donde seguidamente, se describen y justifican los servicios que confeccionarán este sistema de hosting.

A continuación, se explica el funcionamiento de este sistema. Por una parte, se muestra el funcionamiento interno y, por otra parte, se muestra el funcionamiento desde el punto de vista del usuario. Después de ver el funcionamiento, se exponen los resultados obtenidos en las pruebas de rendimiento, seguido de las conclusiones extraídas a lo largo del proyecto.

Finalmente, y sin ser menos importante, se hace una aportación sobre el trabajo futuro, citando las mejoras y extensiones, que este sistema de hosting puede abastecer.

- *E-mail de contacto: enric.torrente@e-campus.uab.cat*
- *Mención realizada: Tecnologias de la Información.*
- *Trabajo tutorizado por: Juan Carlos Sebastián Pérez (Ingeniería de la Información y de las Comunicaciones)*
- *Curso 2017/18*

2 Estado del arte

En la actualidad, numerosas empresas se dedican a la prestación de servicios de alojamiento web y registro de dominios. Estos servicios de alojamiento, según sus características, se reúnen en cuatro grupos [1], [2].

El primero de ellos, el hosting compartido, se caracteriza por ser el más económico, aunque por contrapartida, es el tipo de alojamiento que menos prestaciones y flexibilidad aporta ya que se comparten los recursos del sistema entre todos los usuarios y los servicios vienen predefinidos en la contratación del mismo.

En segundo lugar, tenemos el hosting privado virtual (VPS). En este hosting se comparten los recursos, pero se asegura que cada cliente cuente con los recursos contratados. También cuenta con una preconfiguración de los servicios que el cliente requiera. Estos motivos hacen que el VPS adquiera mayor flexibilidad y prestaciones que el hosting compartido, pero a un precio mayor.

El tercer tipo de hosting, denominado hosting dedicado, es similar al anterior, aunque en este caso no se comparte ningún recurso. El cliente contrata un servidor completo a medida, con la totalidad de sus recursos. El hosting dedicado es más caro que el hosting privado virtual, pero también aporta más flexibilidad y prestaciones ya que permite instalar cualquier servicio o incluso el sistema operativo que el cliente requiera.

En último lugar, tenemos el hosting en la nube. Este tipo de hosting, puede adquirir las filosofías del hosting compartido, el hosting privado virtual o el hosting dedicado. La característica fundamental del hosting en la nube se basa en la distribución de los recursos en varios centros de datos situados en distintas ubicaciones. El hecho de no depender de un único centro de datos ayuda a aumentar el tiempo de actividad por fallos en el hardware.

Actualmente, las empresas dedicadas a ofrecer servicios de hosting utilizan múltiples tecnologías para facilitar el despliegue de los servicios y aportar mayor disponibilidad y rendimiento a sus sistemas.

Para facilitar el despliegue de los servicios, hay empresas que utilizan tecnologías de contenedores de software como Docker, Chuses OpenShift o Kuberneteso, como la empresa Arsys [3], o empresas como SiteGround [4] que utilizan contenedores Linux (LXC).

El uso de clústeres de servidores, como los que utiliza la empresa Freehostia [5] en sus servicios de hosting compartido, ayuda a balancear la carga y aumentar la disponibilidad del sistema.

También se puede combinar el uso de diferentes tecnologías para consolidar varios aspectos del sistema. Si se quiere conseguir una mayor disponibilidad y a su vez un fácil despliegue del servicio, se puede combinar, el uso de contenedores dentro de un clúster de servidores con la distribución de recursos en varios centros de datos.

Otro punto fundamental sobre los servicios de alojamiento es el panel de control para la administración del hosting. Algunas empresas optan por el desarrollo propio del panel de control, en cambio, otras, deciden implementar paneles de control existentes bajo licencias gratuitas o licencias de pago [6].

3 Objetivos

El objetivo principal del proyecto consiste en el diseño e implementación de un sistema para ofrecer servicios de alojamiento web y de bases de datos (BBDD) compartidos, teniendo en cuenta la disponibilidad y escalabilidad de dicho sistema. Para el alojamiento web, se utilizará como base el servicio *HTTP de Apache* y para el servicio de BBDD se utilizará *MySQL*. Este servicio, bautizado como *UAB Hosting*, debe ser accesible a los usuarios que lo requieran a través de un aplicativo web, el cual permitirá la gestión del alojamiento.

Para ayudar a satisfacer las propiedades de escalabilidad y disponibilidad que el objetivo principal requiere, se utilizarán tecnologías de clusterización y monitorización proactiva. También se hará uso de las tecnologías *PHP*, *JavaScript* y *HTML* para el desarrollo del panel de control del hosting junto con el desarrollo de scripts *BashShell* para la ejecución de procesos automatizados de configuración del hosting.

4 Metodología

Una vez definidos los objetivos del proyecto, se observa que éste abarca varias competencias del ámbito de la informática como la administración y gestión de redes, administración de sistemas y servicios o la programación de aplicaciones web. Antes de empezar con el desarrollo del sistema de hosting compartido, se han buscado fuentes de información para comprender, evaluar y escoger las diferentes tecnologías que darán forma a este servicio. Una vez revisadas las fuentes propicias y analizados los requerimientos del sistema, empleando quince horas para dichas tareas, se ha realizado una planificación, tal y como se muestra en el apéndice A1, para llevar a cabo la implementación del proyecto.

Por la naturaleza del proyecto, la planificación está dividida en cinco fases.

En la primera fase se ha diseñado la infraestructura del sistema y la red. Para llevar a cabo esta tarea se han empleado cinco horas.

En la segunda fase, se ha configurado el entorno de trabajo implementando el diseño de la infraestructura de la fase anterior, empleando para ello, seis horas.

Para la tercera fase, se han utilizado cincuenta y seis horas. Esta fase consiste en el despliegue y configuración de los servicios necesarios del sistema. Acto seguido, se han realizado las pruebas de funcionamiento de los servicios anteriormente establecidos.

En la cuarta fase, se ha llevado a cabo la creación del panel de control para la administración del hosting por parte de los usuarios. Paralelamente se han desarrollado los scripts que interactuarán con el sistema en función de las acciones que el usuario realice a través del aplicativo web. Acto seguido, se ha verificado el funcionamiento del aplicativo web y la integración de los scripts. Esta fase ha ocupado gran parte del proyecto, con sus ciento setenta y cinco horas.

En la quinta fase, se han realizado pruebas del funcionamiento del sistema y seguidamente se han efectuado test de rendimiento de los servicios de Apache y MySQL.

Adicionalmente, se ha cumplimentado una tarea marcada como opcional en la planificación inicial del proyecto, la implementación del registro de accesos web mediante AWStats.

5 INFRAESTRUCTURA

La infraestructura que requiere el sistema de alojamiento web ha sido diseñada para satisfacer los objetivos del proyecto. Al no disponer de los recursos hardware necesarios para la puesta en marcha en un entorno real, se ha optado por la virtualización mediante VMWare Workstation instalando en una máquina virtual el hipervisor ESXi 6.5. De este modo se consigue la implementación del sistema mediante hardware simulado por software. La instalación de ESXi en un entorno real se realizaria en un servidor físico. Este paso previo, instalar ESXi como máquina virtual, permite que la instalación posterior de la infraestructura para los servicios de alojamiento web sea similar a la que requeriría la implementación en un entorno real, exceptuando las conexiones de red que pasarían por uno o varios switches de capa tres compatibles con el protocolo 802.1Q [7], pudiendo etiquetar así, la salida de tráfico de las diferentes redes de área local virtuales (VLAN). Los switches o switch, tendrían asignados ciertos puertos para cada VLAN y otro puerto para realizar un enlace troncal (trunk) entre los switches y el firewall. De este modo se consigue mantener el tráfico separado entre VLANs.

La base de esta infraestructura está compuesta de máquinas Linux Debian 9.2. Cada una de estas máquinas tiene configurados los servicios necesarios en función del rol desempeñado en el sistema y están conectadas entre ellas a través de un switch virtual (vSwitch) gestionado por ESXi 6.5 [8]. También hay un *firewall* con el sistema *pfSense* para interconectar las VLAN mediante alias de IP [9] y hacer de puerta de enlace al exterior de la red.

La parametrización de red en cada una de estas máquinas se ha configurado, en base a su acceso, mediante VLANs. Esta configuración facilita la gestión del tráfico en los firewalls y switches. Tal y como se muestra en la Fig.1, las máquinas accesibles desde el exterior, *ams1*, *ns1*, *ns2*, *has1*, *has2* y *ftps1*, pertenecen a la VLAN 10.0.2.0/24, y las no accesibles desde el exterior, *nfs1*, *aps1*, *aps2*, *myds1*, *myds2*, *mghas1* y *mghas2*, pertenecen a la VLAN 10.0.3.0/24. También hay una tercera VLAN, 10.0.100.0/24, para uso exclusivo de monitorización, formada por la máquina *mons*.

6 SERVICIOS

Para ofrecer las prestaciones que el sistema requiere, son necesarios múltiples servicios, que adyacentes entre ellos, confeccionan el sistema de hosting.

Antes de empezar con la instalación y configuración de los servicios, se han configurado todas las máquinas del sistema con su respectivo nombre seguido del dominio *uabhosting.cat* y con su configuración IP correspondiente. Para ello se han modificado los archivos */etc/hostname*, */etc/resolv.conf* y */etc/network/interfaces*. En el apéndice A2 se muestra la configuración de estos archivos para el servidor *ams1*.

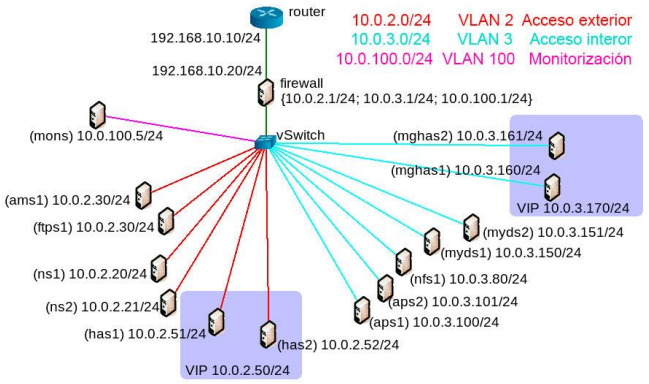


Fig. 1. Diseño de la infraestructura del sistema de hosting UAB Hosting. Las máquinas se describen con el nombre entre paréntesis y la IP asociada. Las IP virtuales (VIP) se asocian al conjunto servidores sombreados.

Las máquinas *ns1* y *ns2* desempeñan el papel de Sistema de Resolución de Nombres (DNS), siendo *ns1* el servidor DNS maestro y *ns2* el servidor DNS esclavo. Estos servidores DNS se han implementado con Berkeley Internet Name Domain (BIND) DNS [10] en la versión *BIND9.10.3*. Este servicio es de código abierto, cumple con todos los estándares del protocolo DNS y se puede configurar como autoritativo, recursivo o ambos. También se han analizado otros servicios DNS como *PowerDNS* y *Unbound* y se observa que *BIND9* ofrece un término medio de respuesta entre estos dos, puesto que *PowerDNS* intenta responder rápido o enviar una respuesta fallida y *Unbound* opta por una respuesta válida, aunque tarde más tiempo de lo habitual [11].

Para configurar el servicio DNS [39, p. 197], en primer lugar, se ha modificado el archivo */etc/bind/named.conf.local*, añadiendo la zona de busqueda directa e inversa con el dominio principal *uabhosting*.cat. Seguidamente se han creado los archivo de datos con los registros de búsqueda directa e inversa del dominio *uabhosting.cat*, correspondientes a la infraestructura del sistema. El archivo de búsqueda directa se encuentra en */etc/bind/local-zones/uabhostingcat.db* y el de búsqueda inversa en */etc/bind/local-zones/uabhostingcat.dbinv*. También se ha modificado el archivo */etc/bind/named.conf.options* para habilitar el servicio DNS como recursivo y autoritativo, [12] permitiendo el acceso al DNS esclavo para que éste también pueda resolver los nombres del mismo modo que lo hace el DNS maestro.

Para la configuración de los dominios regsitrados en el sistema, se ha creado un archivo que se modifica dinámicamente mediante un script. Este archivo está ubicado en */etc/bind/named.conf.uabhosting-zones* y los datos de zona residen en los archivos con extensión *db* dentro del directorio */etc/bind/zones/*.

Todas las configuraciones de los archivos del servicio DNS se muestran en el apéndice A3.

El servidor *ftps1* dispone del servicio de Protocolo de Transferencia de Archivos (FTP) implementado con

PureFTPd [13] y configurado para añadir cifrado de seguridad en la capa de transporte (TLS) [14]. Se ha optado por el uso de este programa ya que incorpora acceso mediante usuarios virtuales y gestión de cuotas de almacenamiento en un único servicio. También cabe mencionar que *PureFTPd* es un programa gratuito de código abierto. Para llevar a cabo la implementación de este servicio se ha utilizado la versión *1.0.43* de *PureFTPd*.

Para la configuración del servicio FTP se ha creado el usuario *ftpuser* en el servidor *ftps1* y se ha añadido al grupo *ftpgroup*. El usuario *ftpuser* no tiene acceso para ejecutar programas en el sistema, únicamente se utiliza para asociar a los usuarios virtuales de *PureFTPd* y sus permisos a un usuario existente en el sistema. Para la gestión de permisos en el servidor *nfs1*, donde se almacenan los archivos del usuario, también se ha creado el grupo *ftpgroup*. Los usuarios virtuales de *PureFTPd* se almacenan en el archivo de datos */etc/pure-ftpd/pureftpd.passwd* donde también s guarda la contraseña cifrada y la raíz del directorio FTP de cada usuario virtual. Este servicio de FTP también genera el archivo oculto *.ftpquota* en el directorio raíz de cada usuario virtual que disponga de una cuota de almacenamiento.

Por otra parte, en el servidor *ams1*, reside el aplicativo web con el panel de control para la gestión del alojamiento, siendo ésta la única vía de acceso para la configuración de los dominios y del hosting desde el punto de vista del usuario. Este servidor tiene instalados los servicios de Apache, PHP y MySQL para servir la aplicación web del panel de control del hosting y almacenar los datos y configuraciones de los usuarios registrados en el sistema.

La función de los servidores *has1* y *has2* constituye el front-end a modo de proxy con funciones de balanceo de carga y conmutación por error para los servidores *aps1* y *aps2*, donde residen los servicios HTTP de Apache. Si uno de los dos servidores *has*# y *aps*# deja de funcionar, el servicio web de Apache se mantiene activo. Para ello se han configurado los servicios de *HAProxy 1.7.5* y *KeepAlived 1.3.2* en los servidores *has*#.

El servicio *HAProxy* [15] permite la redirección del tráfico a los servidores denominados como back-end, en este caso los servidores de Apache *aps*#. En el archivo de configuración de *HAProxy*, se han configurado los back-end con los puertos *80* para peticiones *HTTP* [39, p. 158] y *443* para peticiones *HTTPS* [16]. También se ha configurado la persistencia de las sesiones *PHP* [17], ya que al haber servidores intermedios entre el servicio de Apache y el navegador del cliente, las sesiones no se mantienen si no se han configurado devidamente los servidores *HAProxy*. En el apéndice A4 se muestra la parametrización del archivo de configuración de *HAProxy* ubicado en */etc/haproxy/haproxy.cfg*.

El servicio *KeepAlived* [18] asigna un servidor *has*# como *master* mediante una IP virtual (VIP) sincronizada. Esta asignación se realiza mediante una prioridad que disminuye o incrementa en función del estado de *HAProxy*, si el servicio responde se mantiene la prioridad, si por el contrario, el servicio no responde, se decrementa la prioridad y se asigna al otro servidor *has*# el rol de *master*. Para la asignación de la VIP entre los servidores *has*#, se utiliza el protocolo de redundancia de enrutador virtual (VRRP) [19], definido en el RFC2338 [20]. Para ello se asigna un identificador de *enrutador virtual* que se define en el archivo de configuración de *KeepAlived* [39, p. 162] ubicado en */etc/keepalived/keepalived.conf*. Este identificador debe ser el mismo en ambos servidores *has*#. También se utiliza un script ubicado en */usr/local/bin/check_haproxy.sh* para realizar la comprobación del servicio *HAProxy*. La configuración relativa al servicio de *KeepAlived* y el script de comprobación del servicio se muestra en el apéndice A5.

El servidor *nfs1* cuenta con un sistema de archivos en red (NFS) [21] para que los servidores *ftps1*, *aps1* y *aps2*, mantengan los datos sincronizados. En el apéndice A6 se muestra la parametrización del archvio */etc/exports* para la configuración del servidor NFS. Este servicio es el más crítico de la infraestructura puesto que no se ha formado un clúster de datos por indisponibilidad de recursos hardware.

Para montar el sistema de archivos NFS en los servidores *ftps1*, *aps1* y *aps2* al iniciar el sistema es necesario la instalación del cliente NFS y configurar el archivo */etc/fstab*, tal y como se muestra en el apéndice A7.

Las máquinas *myds1* y *myds2*, junto con *mghas1* y *mghas2*, forman un clúster de BBDD MySQL en red (MySQL NDB Cluster). Por una parte, tenemos los servidores *myds*#, encargados del almacenamiento de las BBDD y del servicio motor de Lenguaje de Consultas Estructuradas (SQL). Por otra parte, tenemos los servidores *mghas*#, encargados de la sincronización y administración del clúster. Los servidores *mghas*# también incorporan el servicio *HAProxy* para repartir la carga de peticiones entre los dos servidores *myds*# y *KeepAlived* para acceder a dichos servidores a través de una VIP tal y como se ha mencionado en los servidores *has*#. En la Fig. 2 se muestra la estructura de *MySQL NDB Cluster* que incorpora los servicios *HAProxy* y *KeepAlived*.

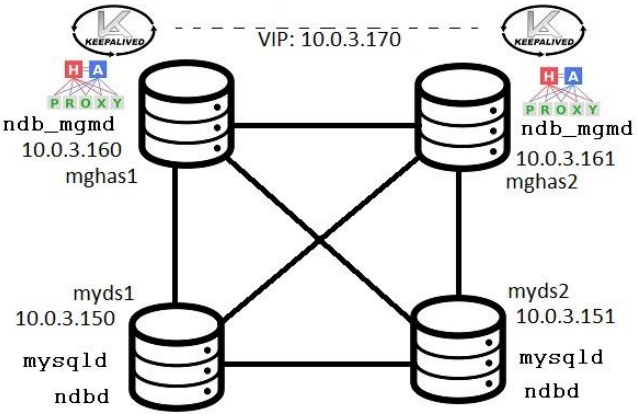


Fig. 2. Estructura de la interconexión del clúster MySQL NDB Cluster.

Los usuarios acceden a MySQL a través de los servidores *aps*# mediante el aplicativo web *phpMyAdmin* de forma transparente que apunta a la VIP 10.0.3.170. Esta IP virtual está asociada a uno de los dos servidores *mghas*#. El servidor *mghas*# redirige la petición a uno de los dos servidores *myds*# que incorporan el servicio *mysqld* y las bases de

datos que se sincronizan mediante el servicio *ndbd*.

Si uno de los dos servidores de administración, *mghas*#, y uno de los servidores de datos y motor sql, *myds*#, deja de funcionar, el servicio de BBDD MySQL se mantiene activo, tal y como se muestra en el apéndice A8.

La versión utilizada en *MySQL NDB Cluster* es la *7.5.8* y los servicios *MySQL* cuentan con la versión *5.7.20*. Esta versión de *MySQL NDB Cluster* es la última versión estable a fecha actual para Debian 9 y su configuración no está documentada totalmente por el fabricante, de tal modo que se han consultado varias fuentes [22], [23], [24], [25] para consolidar su configuración y comprobar el correcto funcionamiento del clúster.

Para que los permisos de usuario se sincronicen entre ambos nodos *ndbd*, es necesario ejecutar un script que proporciona el fabricante bajo el nombre *ndb_dist_priv.sql*. También se han tenido que crear y añadir manualmente los scripts de inicio y parada de los servicios al iniciar y detener el sistema. En el apéndice A9 se muestran los procesos de instalación y configuración para los diferentes nodos, junto con el archivo de configuración del servicio *HAProxy* para el clúster *MySQL*.

Por último, en el servidor *mons*, recaen las funciones de monitorización proactiva mediante el servicio de *Zabbix Server 3.4.3*. Todas las máquinas y servicios del sistema, junto con la red, están monitorizados con este sistema, minimizando así, riesgos de pérdidas en el servicio. Se ha optado por el uso de *Zabbix* ya que ofrece recursos para la monitorización de servicios, estado de las máquinas y red, frente a otros sistemas de monitorización que únicamente se ajustan a la monitorización de servicios o redes [26].

Para la puesta en marcha de *Zabbix* [27], previamente se ha instalado *Apache* para la interfaz de administración web y *MySQL* para almacenar la información de monitorización. Después se ha instalado el servicio de *Zabbix*, en el servidor *mons* y por último los agentes de *Zabbix* en todos los servidores de la infraestructura, incluido el servidor *mons* y el *firewall*. Una vez instalados los agentes, se ha configurado el servidor *Zabbix* para monitorizar dichos servidores con sus respectivos servicios. Para ello, *Zabbix* utiliza plantillas en función del tipo de servicio o servidor a monitorizar. También se pueden definir manualmente los valores de monitorización. Este servicio de monitorización gratuito también cuenta con mapas de infraestructura, gráficos y tablas para la visualización de la información de los activos monitorizados. En el apéndice A10 se muestra un fragmento de la infraestructura del sistema monitorizado a partir de la creación de un mapa mostrando los valores de uso de memoria principal, procesador y la entrada y salida de tráfico de red.

7 FUNCIONAMIENTO

El servicio de hosting que nos ocupa, adquiere dos perspectivas en cuanto a funcionamiento. Por una parte, existe el funcionamiento del sistema no visible al usuario y, por otra parte, el funcionamiento del panel de control, al cual tiene acceso el usuario. Estas dos perspectivas conviven entre ellas para consumar el servicio final de hosting.

7.1 Back-end

El funcionamiento de la parte interna del sistema, se compone de varios procesos

Cuando se registra un dominio en el sistema, el back-end del panel de control, añade el dominio, los usuarios de acceso de MySQL, AWStats y FTP, junto con el resto de información en la BBDD. Seguidamente se ejecutan diferentes scripts en los servidores *ns1*, *nfs1*, *ftps1* y *ams1* mediante la función *ssh2_exec* de PHP [28], [29].

En el servidor DNS maestro, *ns1* se ejecuta el script encargado de crear la nueva zona DNS con los parámetros por defecto. Una tarea cron que se ejecuta cada quince minutos, sincroniza los cambios de zonas. En el apéndice A11 se muestra un fragmento del script de configuración de los DNS y el script que se ejecuta mediante cron.

En el clúster *MySQL NDB*, se registra el usuario de acceso a las BBDD y se añade al grupo *uabhostingusuarios* de la BBDD *phpmyadmin* con el objetivo de mostrar únicamente las opciones necesarias del aplicativo *phpMyAdmin*.

A continuación, se ejecuta el script *virtualhost.sh* en el servidor *nfs1* que se encarga de la creación del hosting virtual de Apache junto con la creación de la estructura de directorios necesarios para este propósito. Después, se crea y configura el acceso al sitio de estadísticas de las visitas de *AWStats* [30] mediante el script *awstats*.sh. Para impedir el acceso el sitio de *AWStats* de cada dominio, se ha configurado el acceso mediante autentificación de usuario [31], con el archivo *.htpasswd*. Después de este proceso, para que la configuración tenga efecto se programa una tarea cron que recargará la configuración de los servidores Apache *aps1* y *aps2*, tal y como se muestra en el apéndice A12.

Una vez creado el sitio *AWStats*, se programa una tarea cron para actualizar los registros de acceso de *aps1* y *aps2*, tal y como se muestra en el apéndice A13. Estos registros están separados en dos archivos identificados con el nombre de la máquina y dominio, ya que *aps1* y *aps2* son máquinas distintas. Para unir los registros en un solo archivo, *AWStats* proporciona un script en Perl denominado *logresovemerge.pl*.

Finalmente, se crea el usuario de acceso FTP mediante un script en Bash ejecutado en el servidor *ftps1*, tal y como se muestra en el apéndice A14.

Para la ejecución de los scripts remotamente se utiliza el protocolo Secure SHell (SSH) mediante claves Rivest, Shamir y Adleman (RSA) [32] para conectar directamente desde los servidores habilitados que contengan las mismas claves, evitando así la introducción de la contraseña.

Una vez registrado el dominio, cuando se accede desde el navegador para acceder al sitio web, el dominio apunta a la VIP de los servidores *has*# y en función de la petición, HTTP o HTTPS, redirige al puerto *80* o *443* respectivamente de los servidores *aps*# comprobando que el servicio Apache esté disponible y utilizando el método Round-Robin para redirigir la petición al servidor *aps1* o *aps2*.

Para modificar el protocolo de Apache de HTTP a HTTPS [33] en los servidores aps#, se añade el enlace simbólico del virtualhost con el protocolo correspondiente y se elimina el opuesto en *sites-enabled*, tal y como se muestra en el apéndice A15. Por otra parte, en el apéndice A16, se

muestra un fragmento del script que permite la redirección, ya sea permanente o temporal, a otro destino.

Paralelamente, el servicio de monitorización, recaba los valores de todos los activos del sistema, alertando de las incidencias cuando estos valores salen del rango estipulado.

También se realizan copias de seguridad [34] de contenido y BBDD diariamente, mediante un script programado en una tarea cron, manteniendo dichas copias de seguridad durante un periodo de treinta días.

7.2 Front-end

Cuando el usuario accede al servicio de hosting, denominado UAB Hosting, inicialmente se solicita el registro del dominio. Una vez registrado el dominio, se accede al panel de control. Para configurar el alojamiento, se disponen de siete opciones generales, tal y como se muestra en el apéndice A17:

- Configuración DNS, donde se pueden crear, modificar y eliminar registros de tipo *A* y *CNAME*.
- Configuración del dominio y subdominio, pudiendo crear y eliminar subdominios y modificar el protocolo HTTP y HTTPS de los dominios y subdominios.
- Acceso FTP, donde se facilita la dirección de acceso y el usuario y contraseña, generados de forma seudoaleatoria.
- Acceso MySQL, donde se muestra la contraseña y usuario junto con la dirección URL de acceso a *phpMyAdmin*. También existe la opción de crear y eliminar la BBDD MySQL.
- Acceso AWStats, donde se dispone del usuario y contraseña para acceder a las estadísticas de acceso web junto a la dirección URL de acceso.
- Acceso a restauración de copias de seguridad. En esta opción se muestra un mensaje informativo sobre el procedimiento para la restauración de las copias de seguridad de contenido y BBDD de los últimos treinta días.
- Acceso a la información del hosting, donde se detalla el espacio consumido sobre los 100 MB de límite. También permite la modificación de contraseña y correo electrónico del usuario.

La BBDD relacional de la Fig. 3 que utiliza el aplicativo web para la gestión de los dominios y alojamientos se ha diseñado con el objetivo de almacenar los datos de usuario y configuraciones.

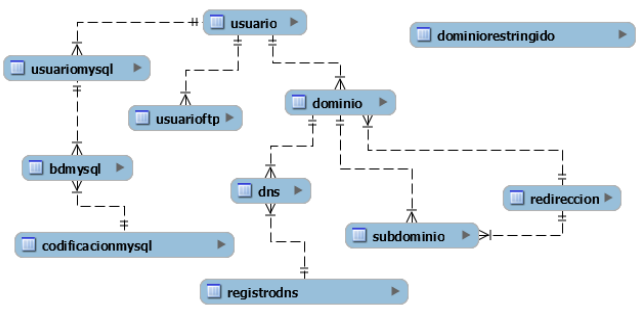


Fig. 3. Diseño de la BBDD relacional utilizada por el aplicativo web para la gestión de los usuarios y configuración del hosting.

Las tablas de esta BBDD se relacionan mediante identificadores que se usan como claves primarias (PK) y claves foráneas (FK), que, junto con el uso de transacciones, ayudan a mantener la integridad de los datos.

Por otra parte, se ha creado una tabla para evitar el registro de los dominios insertados en esta tabla.

Para desarrollar el panel de control se ha utilizado *PHP* para el back-end y *HTML* y *JavaScript* junto con el framework de código abierto *Bootstrap* para el front-end. Para la página web se ha usado la técnica Modelo Vista Controlador (MVC) para separar los datos, la lógica y la interfaz del usuario. También se ha usado una plantilla base [35] con estilos preconfigurados, aunque posteriormente se han adaptado a las necesidades del proyecto que nos ocupa.

8 Resultados

Se han realizado pruebas de rendimiento de los servicios de Apache y MySQL.

Se han contrastado los resultados entre el servicio Apache sin clusterizar, obtenidos del servidor *ams1*, y el servicio Apache clusterizado, obtenidos del conjunto *has1*, *has2*, *aps1* y *aps2*. También se ha aplicado esta metodología al servicio de MySQL, contrastando los resultados obtenidos de *ams1* con los del clúster MySQL NDB formado por los servidores *mghas1*, *mghas2*, *myds1* y *myds2*.

Para realizar las pruebas se han asignado los mismos recursos virtuales, administrados por el hipervisor ESXi 6.5, a las máquinas *ams1*, *has1*, *has2*, *aps1* y *aps2*, siendo 2 GB de memoria principal y dos procesadores Intel Core i5-4300U con dos núcleos cada uno. Para evitar retrasos por el procesamiento del tráfico en el firewall, se han ejecutado las pruebas desde una máquina perteneciente a la misma VLAN en cada caso. De este modo el tráfico pasa únicamente por el vSwitch antes de llegar a su destino.

8.1 Servicio HTTP de Apache

Para las pruebas de rendimiento de Apache se ha utilizado ApacheBench. [36] Los resultados obtenidos muestran el tiempo de respuesta en función del número de peticiones.

Los resultados obtenidos del servicio Apache sin clusterizar muestran menor tiempo de respuesta en peticiones no concurrentes o menos de mil peticiones concurrentes.

Por el contrario, el servicio Apache clusterizado, muestra tiempos más bajos de respuesta cuando el número de peticiones simultaneas va de mil a diez mil, según las pruebas realizadas, mostradas en lo Fig. 4.

En el servicio clusterizado de Aapche se observa que el tiempo de las primeras peticiones es más elevado que en el servicio sin clusterizar. Por otra parte, también se observa que, a partir de tres mil trescientas peticiones concurrentes, el servicio de Apache sin clusterizar, deja de responder peticiones, en cambio el servicio Apache clusterizado responde correctamente a las diez mil peticiones.

8.2 Servicio MySQL

En el servicio MySQL se ha utilizado la herramienta MySQLSlap [37], facilitada por el propio fabricante, para realizar los test de rendimiento.

Se han realizado dos pruebas con distintos tipos de datos. En el primer test, se ha considerado una columna con datos de tipo *int* y otra columna con datos de tipo *char y e*n el segundo test se han usado cinco columnas de tipo *int* y cinco columnas de tipo *char* para realizar la consulta.

En la tabla 1 se muestran los resultados obtenidos en la primera prueba, donde se observa un mayor rendimiento en el servicio MySQL clusterizado con cien conexiones simultaneas con un rango de dos a cincuenta consultas por conexión.

En los resultados del segundo test, mostrados en la tabla 2, se observa un mayor rendimiento en el servicio MySQL clusterizado en cincuenta y cien conexiones simultaneas, estableciendo por cada conexión, entre dos y cincuenta consultas.

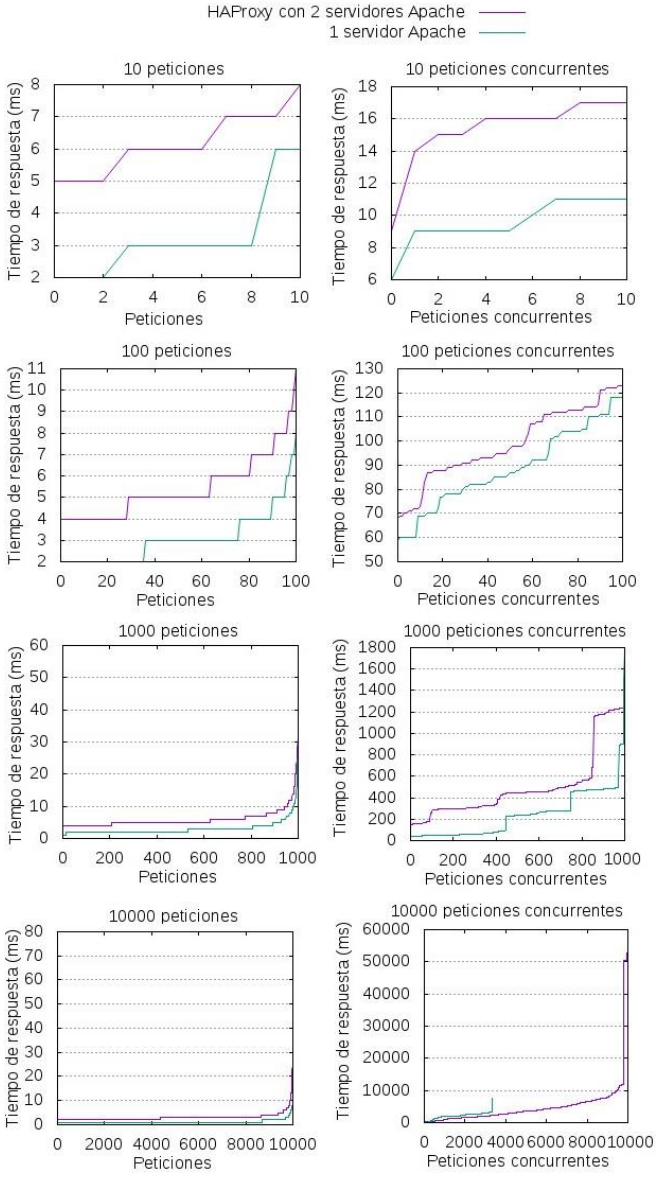


Fig. 4. Gráficos que muestran los resultados de los test del servicio Apache clusterizado, formado por un servidor *has#*, *aps1* y *aps2*, y el servicio Apache sin clusterizar del servidor *ams1*. Se han usado valores de diez, cien, mil y diez mil peticiones, concurrentes y no concurrentes. Para la obtención de resultados, se ha realizado el mismo test cinco veces y se ha calculado la media de los tiempos de respuesta.

TABLA 1

TIEMPO PROMEDIO: 1 CAMPO INT Y 1 CAMPO CHAR

		Conexiones simultáneas									
		2		10		25		50		100	
		MySQL	MySQL NDB	MySQL	MySQL NDB	MySQL	MySQL NDB	MySQL	MySQL NDB	MySQL	MySQL NDB
Consultas	2	0,039	0,071	0,149	0,188	0,230	0,409	0,627	0,999	2,348	2,199
	10	0,048	0,074	0,151	0,218	0,311	0,461	0,851	1,025	2,387	2,251
	25	0,053	0,079	0,155	0,228	0,352	0,465	0,865	1,033	3,452	2,326
	50	0,056	0,080	0,170	0,232	0,355	0,519	0,935	1,050	8,914	2,336

Tiempo promedio en segundos para la ejecución de todas las consultas por cada conexión. Se ha realizado la consulta con un campo de tipo entero y otro de tipo carácter. Para la obtención de resultados, se ha realizado el mismo test cinco veces calculando la media de los tiempos obtenidos.

TABLA 2

TIEMPO PROMEDIO: 5 CAMPOS INT Y 5 CAMPOS CHAR

		Conexiones simultáneas									
		2		10		25		50		100	
		MySQL	MySQL NDB	MySQL	MySQL NDB	MySQL	MySQL NDB	MySQL	MySQL NDB	MySQL	MySQL NDB
Consultas	2	0,067	0,113	0,234	0,356	0,883	0,975	2,385	2,308	7,764	6,870
	10	0,080	0,123	0,354	0,391	0,898	0,996	2,464	2,429	8,514	7,020
	25	0,082	0,127	0,354	0,406	0,973	0,997	2,489	2,458	8,972	7,633
	50	0,090	0,129	0,359	0,425	1,091	1,016	2,666	2,467	16,524	8,515

Tiempo promedio en segundos para la ejecución de todas las consultas por cada conexión. Se ha realizado la consulta con cinco campos de tipo entero y cinco de tipo carácter. Para la obtención de resultados, se ha realizado el mismo test cinco veces calculando la media de los tiempos obtenidos.

9 CONCLUSIONES

En este proyecto se han mostrado todas las etapas necesarias para el desarrollo de un sistema de hosting de alta disponibilidad.

Primero se han explicado los aspectos técnicos de la infraestructura. Después, se han detallado los servicios necesarios para el desarrollo de dicho sistema. Seguidamente, se ha mostrado el funcionamiento del hosting, separando la parte front-end del back-end y finalmente, se han analizado los aspectos de rendimiento más relevantes de este sistema, viendo diferencias significativas en los Servicios Apache y MySQL clusterizados y los no clusterizados. En los resultados obtenidos se muestra un mayor rendimiento de los servicios clusterizados cuando la concurrencia en las peticiones es muy elevada. En cambio, en concurrencias bajas, el rendimiento es mayor en el sistema sin clusterizar. Esto sucede, en gran parte, por el tráfico de red generado adicionalmente, para la gestión de los clústeres en red.

Los resultados indican que el uso de clústeres de red no siempre es bueno en todos los contextos. Se debe analizar el entorno y realizar las previsiones de carga del sistema oportunas para poder llevar a cabo una implementación del servicio acorde con análisis previo.

Por otra parte, también se debe tener en cuenta la disponibilidad del servicio. Con los servicios clusterizados de MySQL y Apache se consigue una mayor disponibilidad frente a los no clusterizados ya que si un servidor del clúster deja de funcionar, el resto de servidores soportan la carga del trabajo, en cambio, en el servicio sin clusterizar,

si el servidor falla, el servicio deja de estar disponible.

La infraestructura del sistema de hosting llevado a cabo en este proyecto, permite afrontar un gran número de peticiones web y consultas sql frente a un sistema con las mismas características, pero sin ser clusterizado, tanto en el servicio Apache como en el servicio MySQL que se ofrece a los usuarios.

Por último, cabe mencionar que una buena planificación ayuda al desarrollo del proyecto, aunque en ocasiones, el tiempo real para determinadas tareas sea mayor del asignado inicialmente, por motivos imprevistos durante el transcurso del proyecto. En nuestro caso, algunas de las tareas realizadas en el proyecto se han llevado a cabo por primera vez, por lo tanto, no se partía de los tiempos de referencia que con la experiencia se adquieren. Por este motivo, se han asignado tiempos con un cierto margen de error a este tipo de tareas.

10 Trabajo futuro

El proyecto UAB Hosting puede adquirir diversidad de mejoras y extensiones del servicio.

Una línea de trabajo futura es la implementación de seguridad en la red y en los servidores, configurando el filtro de seguridad del firewall con regles que definan que de tráfico de entrada y salida está permitido. Otra manera de aumentar la seguridad del sistema, no excluyente de la primera, es la implementación de un servicio de detección de intrusiones en red (NIDS) como Snort [38].

Otra mejora es la integración de varias versiones de PHP, ya que actualmente existe la versión 7 de PHP y en el hosting que se ha desarrollado se ha utilizado la versión 5. Para que el usuario pueda escoger la versión de PHP que más se ajusta a su proyecto, se establecería mediante el panel de control.

Actualmente para la restauración de copias de Seguridad, el usuario necesita contactar con el servicio técnico. Una mejora considerable pasaría por implementar un sistema automático de restauración mediante la interfaz web a la que el usuario tiene acceso.

El sistema NFS para almacenar los datos, también es un candidato a considerar su mejora, para aumentar la disponibilidad del sistema.

El hecho de poder tener varios dominios asignados a un único usuario o varios poder crear usuarios FTP y MySQL también se tienen en consideración en las líneas de trabajo futuras junto con el bloqueo del tamaño de las BBDD de usuario o el bloqueo de ancho de banda por dominio.

Una posible extensión al hosting del proyecto realizado, es la instalación de los principales sistemas de gestión de contenidos (CMS) de forma automática mediante un asistente.

La implantación de servicios de correo también forma parte de las posibles extensiones que puede abarcar este sistema, aunque esta implantación no sería funcional para el sistema actual, puesto que el sistema de hosting UAB Hosting no está certificada ni registrada por la Corporación para la Asignación de Nombres y Números de Internet (ICANN). Esto implica que los dominios registrados únicamente son válidos en un entorno local con los servidores DNS del sistema UAB Hosting para la resolución de nombres. Una alternativa para asociar nombre de dominio registrados mediante una entidad acreditada, es la vinculación de los servidores DNS de esta empresa para la resolución de las IPs donde se alojan los servicios de hosting y los posibles servicios de correo de UAB Hosting.

Agradecimientos

Durante el transcurso del proyecto, algunas personas han sido claves. El apoyo recibido por mi familia más cercana ha sido de gran ayuda para afrontar los momentos menos alegres, cuando las coses no funcionan como esperas. Mi pareja, Carola, que, aun no sabiendo demasiado del campo de la informática, se ha interesado en la evolución del proyecto y me ha sabido transmitir esa motivación especial.

Por último y no menos importante, un gran agradecimiento a mi tutor, Juan Carlos Sebastián Pérez, por su gran implicación en el trabajo, motivándome y animándome durante todo este tiempo. La orientación, el seguimiento y la buena comunicación que ha habido, han sido aspectos fundamentales para hacer posible este proyecto.

Biblioografía

[1] J. Stevens, "Different Types of Web Hosting Explained (Shared, VPS, Dedicated)," *HostingFacts.com*, 02-Oct-2017. [Online]. Available: https://hostingfacts.com/different-types-of-web-hosting/. [Accessed: 15-Ene-2018].

[2] "Tipos de hosting: ¿Cuál le conviene más a tu sitio web?," *HostingExperto*, 29-Mar-2017. [Online]. Available: https://www.hostingexperto.es/tipos-de-hosting/. [Accessed: 15-Ene-2018].

[3] M. León, "OpenShift, Docker y Kubernetes, las tecnologías detrás de Cloud Hosting Gestionado de Arsys," *Blog de arsys.es*, 08-Nov-2017. [Online]. Available: https://www.arsys.es/blog/programacion/openshift-docker-kubernetes-cloud/. [Accessed: 18-Dic-2017].

[4] W. H. SiteGround, "Tecnología para el tiempo de actividad de SiteGround," *SiteGround*. [Online]. Available: https://www.siteground.es/tiempo-de-actividad. [Accessed: 05-Oct-2017].

[5] "Load Balanced Cluster Technology," *Load Balanced Web Hosting Cluster Technology | Freehostia.com*. [Online]. Available: https://www.freehostia.com/technology/. [Accessed: 05-Oct-2017].

[6] "2018's Ultimate Guide to Web Panels: cPanel vs. Plesk vs. Webmin vs. Other Popular Hosting Management Tools," *HostingAdvice.com*, 11-Jan-2018. [Online]. Available: http://www.hostingadvice.com/blog/cpanel-vs-plesk-vs-webpanel/. [Accessed: 18-Ene-2018].

[7] IEEE Computer Society, "IEEE Standard for Local and metropolitan area networks Virtual Bridged Local Area Networks", IEEE Std 802.1Q, mayo, 2005.

[8] Dboeva, "vSphere Standard Switches," *vmware information center*. [Online]. Available: https://docs.vmware.com/en/VMware-vSphere/6.5/com.vmware.vsphere.networking.doc/GUID-350344DE-483A-42ED-B0E2-C811EE927D59.html. [Accessed: 05-Oct-2017].

[9] "What are Virtual IP Addresses," *What are Virtual IP Addresses - pfSense Documentation*. [Online]. Available:

https://doc.pfsense.org/index.php/What_are_Virtual_IP_Addresses#IP_Alias. [Accessed: 08-Oct-2017].
[10] "BIND," *Internet Systems Consortium*. [Online]. Available: https://www.isc.org/downloads/bind/. [Accessed: 04-Oct-2017].
[11] H. Boulakhrif, "Analysis of DNS Resolver Performance Measurements," trabajo fin de máster, System and Network Engineering, University of Amsterdam, Ámsterdam, WX, 2015.
[12] Ethand, "Configure Authoritative Name Server Using BIND on Ubuntu," *ProfitBricks DevOps Central*. [Online]. Available: https://devops.profitbricks.com/tutorials/configure-authoritative-name-server-using-bind-on-ubuntu/#configure-primary-server. [Accessed: 06-Oct-2017].
[13] "Frank Denis (Jedi/Sector One)s random stuff," *Documentation*. [Online]. Available: https://www.pureftpd.org/project/pureftpd/doc. [Accessed: 17-Nov-2017].
[14] Howtoforgecom, "How To Configure PureFTPd To Accept TLS Sessions On Debian Lenny," *Howtoforge*. [Online]. Available: https://www.howtoforge.com/how-to-configure-pureftpd-to-accept-tls-sessions-on-debian-lenny#-configuring-pureftpd. [Accessed: 21-Nov-2017].
[15] "HAProxy," *HAProxy version 1.7.10 - Configuration Manual*. [Online]. Available: http://cbonte.github.io/haproxy-dconv/1.7/configuration.html. [Accessed: 25-Sept-2017].
[16] DigitalOcean, "Contents," *How To Implement SSL Termination With HAProxy on Ubuntu 14.04 | DigitalOcean*, 08-Oct-2016. [Online]. Available: https://www.digitalocean.com/community/tutorials/how-to-implement-ssl-termination-with-haproxy-on-ubuntu-14-04. [Accessed: 09-Nov-2017].
[17] *17.3.1 Configuring HAProxy for Session Persistence*, 29-Aug-2017. [Online]. Available: https://docs.oracle.com/cd/E37670_01/E41138/html/section_jyh_zhz_4r.html. [Accessed: 09-Nov-2017].
[18] "Introduction," *Introduction - Keepalived 1.2.15 documentation*. [Online]. Available: http://www.keepalived.org/doc/introduction.html. [Accessed: 23-Oct-2017].
[19] "Case Study: Failover using VRRP," *Case Study: Failover using VRRP - Keepalived 1.2.15 documentation*. [Online]. Available: http://www.keepalived.org/doc/case_study_failover.html. [Accessed: 23-Oct-2017].
[20] The Internet Society, "Virtual Router Redundancy Protocol," RFC 2338, abril, 1998.
[21] L. Rendek, "How to configure NFS on Debian 9 Stretch Linux," *Linux Tutorials - Learn Linux Configuration*, 05-Jun-2017. [Online]. Available: https://linuxconfig.org/how-to-configure-nfs-on-debian-9-stretch-linux. [Accessed: 15-Oct-2017].
[22] "MySQL 5.7 Reference Manual," *MySQL*. [Online]. Available: https://dev.mysql.com/doc/refman/5.7/en/. [Accessed: 13-Oct-2017].
[23] "MySQL NDB Cluster 7.5 :: 7.15 Distributed MySQL Privileges for NDB Cluster," *MySQL*. [Online]. Available: https://dev.mysql.com/doc/mysql-cluster-excerpt/5.7/en/mysql-cluster-privilege-distribution.html. [Accessed: 13-Oct-2017].
[24] "Montar un cluster de MySQL," *adictosaltrabajo*. [Online]. Available: https://www.adictosaltrabajo.com/tutoriales/mysql-cluster/. [Accessed: 14-Oct-2017].
[25] B. Brandli and M. Dick. "MySQL Cluster Tutorial," presented at O'Reilly MySQL Conference & Expo 2010, Santa Clara, California, 2010.
[26] RajuVadicharla et al., "A Open Source Tools & Comparative Study on Cloud Computing". *International Journal of Engineering Research and Development*, vol. 6, pp. 69-73, Apr. 2013.
[27] "Zabbix Documentation 3.4," *Zabbix Manual [Zabbix Documentation 3.4]*. [Online]. Available: https://www.zabbix.com/documentation/3.4/manual. [Accessed: 12-Dic-2017].
[28] PhilKershaw, "Simple code for executing commands on a remote Linux server via SSH in PHP," *Gist*. [Online]. Available: https://gist.github.com/PhilKershaw/1389041. [Accessed: 23-Nov-2017].
[29] "Instalar SSH2 para PHP en Debian o Ubuntu," *ImperioWeb.net*. [Online]. Available: https://www.imperioweb.net/instalar-ssh2-para-php-en-debian-o-ubuntu/. [Accessed: 23-Nov-2017].
[30] "AWStats Installation, Configuration and Reporting," *AWStats Documentation - Setup page*. [Online]. Available: http://www.awstats.org/docs/awstats_setup.html. [Accessed: 27-Dic-2017].
[31] "Autenticacion y Autorizacion," *Autenticación y Autorización - Servidor Apache HTTP Versin 2.4*. [Online]. Available: https://httpd.apache.org/docs/current/es/howto/auth.html#gettingitworking. [Accessed: 09-Ene-2018].
[32] Howtoforgecom, "How To Set Up SSH With Public-Key Authentication On Debian Etch," *Howtoforge*. [Online]. Available: https://www.howtoforge.com/set-up-ssh-with-public-key-authentication-debian-etch. [Accessed: 16-Nov-2017].
[33] DigitalOcean, "Contents," *How To Create a Self-Signed SSL Certificate for Apache in Ubuntu 16.04 | DigitalOcean*, 13-Oct-2016. [Online]. Available: https://www.digitalocean.com/community/tutorials/how-to-create-a-self-signed-ssl-certificate-for-apache-in-ubuntu-16-04. [Accessed: 20-Dic-2017].
[34] Simon Davies, "Backing up MySQL with an automated BASH script," *Simon Davies*, 26-Oct-2017. [Online]. Available: https://simon-davies.name/bash/backing-up-mysql-databases. [Accessed: 15-Ene-2018].
[35] BlackrockDigital, "BlackrockDigital/startbootstrap-sb-admin," *GitHub*. [Online]. Available: https://github.com/BlackrockDigital/startbootstrap-sb-admin. [Accessed: 07-Nov-2017].
[36] "ab - Apache HTTP server benchmarking tool," *ab - Apache HTTP server benchmarking tool - Apache HTTP Server Version 2.4*. [Online]. Available: https://httpd.apache.org/docs/2.4/programs/ab.html. [Accessed: 08-Ene-2018].
[37] "MySQL 5.7 Reference Manual :: 4.5.9 mysqlslap - Load Emulation Client," *MySQL*. [Online]. Available: https://dev.mysql.com/doc/refman/5.7/en/mysqlslap.html. [Accessed: 11-Ene-2018].
[38] *Snort: Open Source Network Intrusion Prevention*. [Online]. Available: https://www.esecurityplanet.com/network-security/Snort-Open-Source-Network-Intrusion-Prevention-3681296.htm. [Accessed: 22-Ene-2018].
[39] S. Ali, *Practical Linux infrastructure:* Berkeley, CA: Apress, 2015.

APÉNDICE

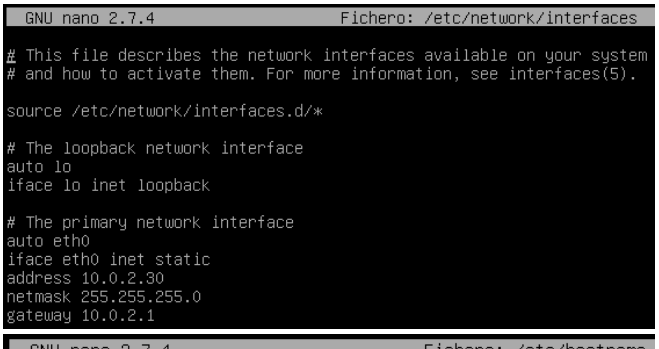
A1. PLANIFICACIÓN

Tabla con la planificación del proyecto UAB Hosting.

Id	Nombre de tarea	Duración	Comienzo	Prede	Fecha límite	% comple
1	**Información, análisis, redacción de documentos y entregas**	**105 horas**	**jue 21/09/17**		**vie 23/02/18**	**81%**
2	Buscar fuentes de información	20 horas	jue 21/09/17		jue 28/12/17	100%
3	Analizar requerimientos del proyecto	3 horas	lun 25/09/17		jue 26/10/17	100%
4	Redactar informe inicial	6 horas	lun 25/09/17	3	dom 08/10/17	100%
...						
15	**Diseño infraestructura y red**	**5 horas**	**lun 25/09/17**		**lun 09/10/17**	**100%**
16	Diseño diagrama de red CORE	4 horas	lun 25/09/17	3	mar 03/10/17	100%
17	Simulación diagrama de red CORE	1 hora	jue 28/09/17	16	lun 09/10/17	100%
18	**Configuración entorno de trabajo**	**6 horas**	**lun 02/10/17**		**jue 12/10/17**	**100%**
19	Instalación entorno virtual ESXi	1 hora	lun 02/10/17	3	lun 09/10/17	100%
20	Instalación máquina virtual Linux Debian (plantilla)	1 hora	lun 02/10/17	3	lun 09/10/17	100%
21	Configuración entorno ESXi	1 hora	mar 03/10/17	19	lun 09/10/17	100%
22	Clonación y configuración de red en las máquinas virtuales en ESXi	3 horas	mié 04/10/17	20;19	jue 12/10/17	100%
23	**Configuración infraestructura de servicios**	**51 horas**	**jue 05/10/17**		**jue 28/12/17**	**100%**
24	Configuración básica firewall	2 horas	jue 05/10/17	22	vie 13/10/17	100%
25	Instalación y configuración DNS BIND	6 horas	lun 09/10/17	22	mar 17/10/17	100%
26	Instalación y configuración Apache + MySQL (administración)	2 horas	mar 10/10/17	22	mar 17/10/17	100%
27	Instalación y configuración Apache	1 hora	mar 10/10/17	22	mié 18/10/17	100%
28	Configuración HTTPS Apache	2 horas	mié 18/10/17	27	jue 28/12/17	100%
29	Instalación y configuración NFS	1 hora	mar 10/10/17	22	mié 18/10/17	100%
30	Instalación y configuración FTP	3 horas	mar 10/10/17	22	mar 12/12/17	100%
31	Instalación y configuración MySQL Cluster	18 horas	mar 10/10/17	22	vie 20/10/17	100%
32	Instalación y configuración 2 HAProxy	10 horas	jue 12/10/17	22	mar 24/10/17	100%
33	Instalación y configuración monitorización ZABBIX	6 horas	dom 15/10/17	22	jue 28/12/17	100%
34	**Página web ALTA usuario y ADMINISTRACIÓN hosting**	**172 horas**	**jue 05/10/17**		**mié 03/01/18**	**100%**
35	Diseño BBDD MySQL (administración)	5 horas	mié 08/11/17		dom 12/11/17	100%
36	Diseño página web	5 horas	mié 08/11/17	35	sáb 18/11/17	100%
37	Programación página web	125 horas	jue 09/11/17	36	lun 25/12/17	100%
38	Integración scripts sistema	20 horas	sáb 18/11/17	35;36	sáb 30/12/17	100%
39	Aplicar estilos	10 horas	lun 20/11/17	37	mar 26/12/17	100%
40	Test página web	7 horas	dom 10/12/17	39	mié 03/01/18	100%
41	**Pruebas del sistema**	**18 horas**	**vie 20/10/17**		**jue 18/01/18**	**100%**
42	Pruebas del sistema completo	8 horas	lun 25/12/17	23;34	jue 04/01/18	100%
43	Pruebas rendimiento Apache (Apache Benchmark y Siege)	10 horas	vie 29/12/17	42	jue 18/01/18	100%
44	**Tareas opcionales**	**11 horas**	**vie 05/01/18**		**jue 25/01/18**	**100%**
45	~~Configuración de seguridad avanzada en firewall~~	~~7 horas~~	~~vie 05/01/18~~	~~41~~	~~jue 25/01/18~~	~~0%~~
46	Implementación registros acceso web	11 horas	dom 07/01/18	45	jue 25/01/18	100%
47	~~Implementación tráfico web~~	~~10 horas~~	~~mié 10/01/18~~	~~45~~	~~jue 25/01/18~~	~~0%~~

A2. CONFIGURACIÓN DE RED

Configuración de red para el host *ams1*.



```
  GNU nano 2.7.4                  Fichero: /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 10.0.2.30
netmask 255.255.255.0
gateway 10.0.2.1
```

```
  GNU nano 2.7.4                  Fichero: /etc/hostname

ams1
```

```
  GNU nano 2.7.4                  Fichero: /etc/resolv.conf

domain uabhosting.cat
search uabhosting.cat
nameserver 10.0.2.20
nameserver 10.0.2.21
```

A3. CONFIGURACIÓN DNS MAESTRO

Configuración del archivo *named.conf*:

```
include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
// Archivo de dominios registrados UAB Hosting
include "/etc/bind/named.conf.uabhosting-zones";
```

Configuración del archivo *named.conf.options*:

```
options {
   directory "/var/cache/bind";
   // Se permite la transeferencia de zonas a ns2
   allow-transfer {10.0.2.21;};
   // Se notifican cambios en las zonas a ns2
   notify yes;
   also-notify {10.0.2.21;};
   // Si no encuentra el dominio localmente,
   // se busca mediante los servidores DNS externos
   forwarders {
      8.8.8.8; 8.8.4.4; 192.168.10.10;
   };
   allow-recursion {any;};
   recursion yes;
};
```

Configuración del archivo *named.conf.local*:

```
//Busqueda directa uabhosting.cat
zone "uabhosting.cat" {type master; file
   "/etc/bind/local-zones/uabhostingcat.db";};

// Busqueda inversa uabhosting.cat
zone "0.10.in-addr.arpa" {type master; file
   "/etc/bind/local-zones/uabhostingcat.dbinv";};
```

Configuración del archivo *named.conf.uabhosting-zones*:

```
//************************************************//
// IMPORTANTE: ESTE ARCHIVO SE MANIPULA Y SINCRONIZA
// AUTOMATICAMENTE
// Editar el archivo ubicado en
// NS1:/etc/bind/syncdns/master
// Cada zona debe estar en una unica linea
//************************************************//
// Archivo definicion de zonas registradas en UAB Hosting
// ZONAS
//Ejemplo dominio “dominio.com” regsitrado en UAB Hosting
zone "dominio.com" {type master; file
   "/etc/bind/zones/dominiocom.db";};
```

Configuración del archivo *uabhostingcat.db*:

```
$TTL 604800
@ IN SOA ns1.uabhosting.cat. hostmaster.uabhosting.cat. (
                     1          ; Serial
                604800          ; Refresh
                 86400          ; Retry
               2419200          ; Expire
                604800 )        ; Negative Cache TTL
;NS Records
@       IN      NS      ns1.uabhosting.cat.
@       IN      NS      ns2.uabhosting.cat.
;A records
@       IN      A       10.0.2.30
www     IN      A       10.0.2.30
ns1     IN      A       10.0.2.20
ns2     IN      A       10.0.2.21
ams1    IN      A       10.0.2.30
vhas    IN      A       10.0.2.50
has1    IN      A       10.0.2.51
has2    IN      A       10.0.2.52
ftps1   IN      A       10.0.2.90
nfs1    IN      A       10.0.3.80
aps1    IN      A       10.0.3.100
aps2    IN      A       10.0.3.101
myds1   IN      A       10.0.3.150
myds2   IN      A       10.0.3.151
mghas1  IN      A       10.0.3.160
mghas2  IN      A       10.0.3.161
vmysql  IN      A       10.0.3.170
mons    IN      A       10.0.100.5
mysql   IN      A       10.0.2.50
;CNAME Records
*       IN      CNAME   10.0.2.30
```

Configuración del archivo *uabhostingcat.dbinv*:

```
$TTL 604800
@ IN SOA uabhosting.cat. root.uabhosting.cat. (
                     1          ; Serial
                604800          ; Refresh
                 86400          ; Retry
               2419200          ; Expire
                604800 )        ; Negative Cache TTL
```

```
;NS Records
        IN      NS      ns1.uabhosting.cat.
        IN      NS      ns2.uabhosting.cat.
;PTR Records
20.2    IN      PTR     ns1.uabhosting.cat.
21.2    IN      PTR     ns2.uabhosting.cat.
30.2    IN      PTR     ams1.uabhosting.cat.
90.2    IN      PTR     ftps1.uabhosting.cat.
51.2    IN      PTR     has1.uabhosting.cat.
52.2    IN      PTR     has2.uabhosting.cat.
50.2    IN      PTR     vhas.uabhosting.cat.
80.3    IN      PTR     nfs1.uabhosting.cat.
100.3   IN      PTR     aps1.uabhosting.cat.
101.3   IN      PTR     aps2.uabhosting.cat.
150.3   IN      PTR     myds1.uabhosting.cat.
151.3   IN      PTR     myds2.uabhosting.cat.
160.3   IN      PTR     mghas1.uabhosting.cat.
161.3   IN      PTR     mghas2.uabhosting.cat.
170.3   IN      PTR     vmysql.uabhosting.cat.
5.100   IN      PTR     mons.uabhosting.cat.
```

A4. CONFIGURACIÓN HAPROXY

Configuración del archivo *haproxy.cfg* en *has*#:

```
global
  ...
defaults
  ...
// Se define el frontend (haproxy) para puerto 80 (HTTP)
frontend apache
 mode http
 option httplog
 bind 10.0.2.50:80
 reqadd X-Forwarded-Proto:\ http
 reqadd X-Forwarded-Port:\ 80
  // Se asignan el backend (los servidores de apache)
 default_backend backend_apache

// Se define el backend para servicio web por el puerto 80
backend backend_apache
 mode http
 // Se asigna Round Robin como método de balanceo
 balance roundrobin
 option tcp-check
 // Se mantienen las sesiones entre aps1 y aps2
 cookie SERVERID insert nocache
 server aps1 10.0.3.100:80 check cookie S1 check port 80
 server aps2 10.0.3.101:80 check cookie S2 check port 80

// Se define el frontend (haproxy) para puerto 443 (HTTPS)
frontend apache_ssl
 mode http
 option httplog
 // Asignación de certificado (el mismo que aps1 y aps2)
 bind 10.0.2.50:443 ssl crt
        /var/ssl/cert/apache-selfsigned.pem
 reqadd X-Forwarded-Proto:\ https
 reqadd X-Forwarded-Port:\ 443
 rspadd Strict-Transport-Security:\ max-age=15768000
 default_backend backend_apache_ssl

// Se define el backend para servicio web por el puerto 443
backend backend_apache_ssl
 mode http
 option tcp-check
 balance roundrobin
 cookie SERVERID insert nocache
 server aps1 10.0.3.100:443 ssl verify none check cookie
        S1 port 443
 server aps2 10.0.3.101:443 ssl verify none check cookie
        S2 port 443
```

A5. CONFIGURACIÓN KEEPALIVED

Configuración del archivo *keepalived.config*:

```
// Script para la comprobación del servicio HAProxy
// Se comprueba cada 2s, si responde debidamente, se incre-
// menta la prioridad en 2, sino, se mantiene la prioridad.
vrrp_script check_haproxy {
  script "/usr/local/bin/check_haproxy.sh"
  interval 2
  weight 2
}
// Instancia para compartir VIP
vrrp_instance VI_1 {
  state MASTER
  interface eth0
// Se asigna el mismo ID en has1 y has2
  virtual_router_id 51
// Se asigna prioridad 101 en has1 y 100 en has2
  priority 101
  virtual_ipaddress {
     10.0.2.50/24 dev eth0
  }
  track_script {
    check_haproxy
  }
}
```

Script *check_haproxy.sh*:

```
#!/bin/bash
# Comprueba que haproxy se este ejecutando
HAPROXY_STATUS=$(/bin/ps ax | grep -w [h]aproxy)
if [ "$HAPROXY_STATUS" != "" ]
  then
    exit 0
else
    exit 1
fi
```

A6. CONFIGURACIÓN SERVIDOR NFS

Configuración del archivo *exports*:

```
/data/webdata 10.0.2.90/32(rw,no_root_squash,async)
        10.0.3.0/24(rw,no_root_squash,async)
        10.0.2.30/32(ro,no_root_squash,async)
/data/webconfig/sites-available
        10.0.3.0/24(rw,no_root_squash,async)
/data/webconfig/sites-enabled
        10.0.3.0/24(rw,no_root_squash,async)
/data/logs 10.0.3.0/24(rw,no_root_squash,async)
/backup 10.0.2.30/32(rw,no_root_squash,async)
```

A7. CONFIGURACIÓN CLIENTE NFS

Configuración del archivo *fstab* en los host *aps1* y *aps2*:

```
# Añadir las siguientes líneas al final del archivo
10.0.3.80:/data/webdata /var/www nfs rw,sync,hard,intr 0 0
10.0.3.80:/data/webconfig/sites-available
    /etc/apache2/sites-available nfs rw,sync,hard,intr 0 0
10.0.3.80:/data/webconfig/sites-enabled
    /etc/apache2/sites-enabled nfs rw,sync,hard,intr 0 0
10.0.3.80:/data/logs /data/logs nfs rw,sync,hard,intr 0 0
```

Configuración del archivo *fstab* en host *ftps1*:

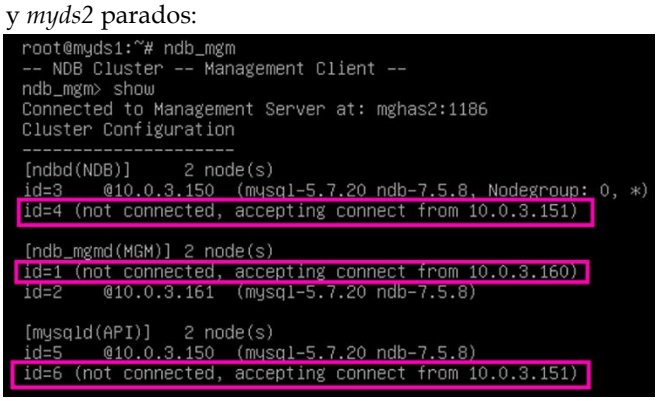
```
# Añadir las siguientes líneas al final del archivo
10.0.3.80:/data/webdata /data nfs rw,sync,hard,intr 0 0
```

Configuración del archivo *fstab* en host *ams1*:

```
# Añadir las siguientes líneas al final del archivo
10.0.3.80:/backup /backup nfs rw,sync,hard,intr 0 0
10.0.3.80:/data/webdata /data nfs ro,sync hard,intr 0 0
```

A8. FUNCIONAMIENTO MYSQL NDB CLUSTER

Funcionamiento del clúster MySQL con los nodos *mghas1* y *myds2* parados:



A9. CONFIGURACIÓN MYSQL NDB CLUSTER

Proceso de instalación MySQL NDB Cluster:

```
#Comun para todos los nodos
apt-get install python-paramiko libclass-methodmaker-perl
wget     https://dev.mysql.com/get/mysql-apt-config_0.8.8-
1_all.deb
apt install gdebi-core
#Escoger la opción MySQL Cluster
gdebi mysql-apt-config_0.8.8-1_all.deb
apt update

# Para nodos SQL (myds1, myds2)
apt install mysql-cluster-community-server

# Para nodos de datos (myds1, myds2)
apt install mysql-cluster-community-data-node

# Para nodos de administración del clúster (mghas1, mghas2)
apt install mysql-cluster-community-management-server
```

Configuración archivo *config.ini* de los servidores *mghas*#:

```
# MySQL CLuster - Data Host (default configuration)
[ndbd default]
DataDir=/usr/local/mysql-cluster/data
DataMemory=1024M
IndexMemory=512M
NoOfReplicas=2

# MySQL Cluster - Management Host
[ndb_mgmd]
NodeId=1
HostName=10.0.3.160
DataDir=/var/lib/mysql-cluster
[ndb_mgmd]
NodeId=2
HostName=10.0.3.161
DataDir=/var/lib/mysql-cluster

# MySQL Cluster - Data Host
[ndbd]
HostName=10.0.3.150
[ndbd]
HostName=10.0.3.151

# MySQL Cluster - SQL Host
[mysqld]
HostName=10.0.3.150
[mysqld]
HostName=10.0.3.151
```

Configuración archivo *mysql.cnf* de los servidores *myds*#:

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/run/mysqld/mysqld.sock
symbolic-links=0
log-error=/var/log/mysql/error.log
pid-file=/var/run/mysqld/mysqld.pid
user=mysql
ndbcluster
port=3307
ndb-connectstring=10.0.3.160,10.0.3.161
default-storage-engine=ndbcluster

[mysql_cluster]
ndb-connectstring=10.0.3.160,10.0.3.161
```

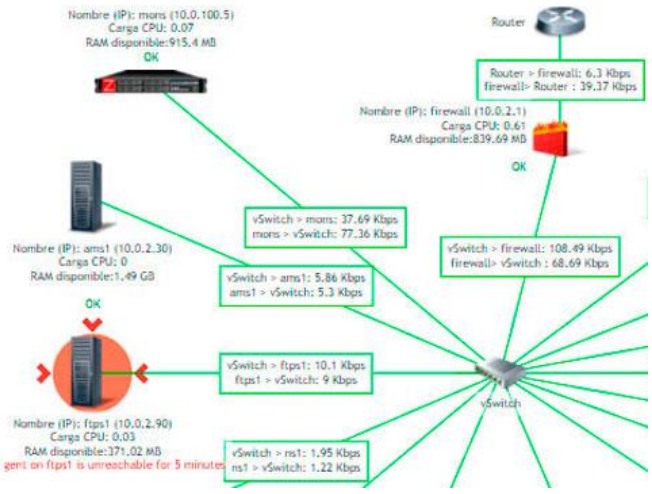
Configuración archivo *haproxy.cfg* en *mghas*#:

```
global
  ...
defaults
  ...
// Se define el frontend (haproxy) para puerto 3306 (MySQL)
frontend mysql-cluster
        bind 10.0.3.170:3306
        mode tcp
        default_backend backend_mysql-cluster

// Se define el backend para MySQL por el puerto 3307
backend backend_mysql-cluster
        balance roundrobin
        option mysql-check user mysql_haproxy_check
        server myds1 10.0.3.150:3307 check
        server myds2 10.0.3.151:3307 check
```

A10. MONITORIZACIÓN ZABBIX

Fragmento del mapa de monitorización de la infraestructura. Se muestra el host *ftps1* con fallo y *mons*, *ams1* y el *firewall* como correcto:



A11. Scripts servicio DNS

Fragmento del script *binddns-config.sh* mostrando la actualización del *Serial* después de un cambio en los registros:

```
#!/bin/bash
...
SN=`grep -e ";Serial" $ZONEDIRFILE | sed -n "s/^\s*\([0-
9]*\)\s*;Serial\s*/\1/p"`
let SN=$SN+1
sed  -i  "s/\t\t\t.*\t;Serial/\t\t\t${SN}\t;Serial/g"  $ZO-
NEDIRFILE
...
```

Script *syncdns.sh*. Este script se ejecuta mediante cron:

```
#!/bin/bash
# Sincroniza los cambios de zonas en el servidor ns1
rsync  -a  /etc/bind/syncdns/master/named.conf.uabhosting-
zones /etc/bind/named.conf.uabhosting-zones

# Sincroniza los cambios de zonas en el servidor ns2 me-
diante SSH
rsync -a /etc/bind/syncdns/slave/named.conf.uabhosting-zo-
nes root@10.0.2.21:/etc/bind/named.conf.uabhosting-zones

# Actualiza los cambios en ns1 sin parar el servicio (reload
del servicio)
service bind9 reload

# Actualiza los cambios en ns2 sin parar el servicio (reload
del servicio) conectado mediante SSH
ssh root@10.0.2.21 service bind9 reload
```

```
root@ns1:~# crontab -l
*/15 * * * * /opt/script/syncdns.sh
```

A12. SCRIPT RECARGA CONFIGURACIÓN APACHE

Script *cronsyncaps.sh* para programar la recarga de la configuración en *aps1* y *aps2*:

```
 #!/bin/bash
apsip=( 10.0.3.100 10.0.3.101 )
for i in "${apsip[@]}"
 do
# Si no existe la tarea cron se programa cada 5m
# Si la tarea cron ya existe no se hace nada
  if [ $(crontab -l | grep -c "/opt/script/syncaps.sh $i")
-eq 0 ]
  then
   (crontab  -u  root  -l  ;  echo  -e  "*/5  *  *  *  *
/opt/script/syncaps.sh $i") | crontab -u root -
  fi
done
exit 0;
```

Script *syncaps.sh* para recargar la configuración en *aps1* y *aps2*:

```
#!/bin/bash
# Se conecta mediante SSH, si se conecta y ejecuta
# correctamente la tarea, se elimina la tarea programada
if ssh root@$1 service apache2 reload
        then
        crontab -u root -l | grep -v "/opt/script/syncaps.sh $1"  | crontab -u root -
fi
```

A13. Unión de registros Apache

Script *synclogs.sh*, utilizado en la unión de los registros Apache de los servidores *aps1* y *aps2* y en la actualización de los registros, ya unidos, para AWStats:

```
#!/bin/bash
dominio=$1
chmod -R 775 /data/logs/apache2/$dominio/
perl     /opt/script/logresolvemerge.pl     /data/logs/apache2/$dominio/*.log    >    /data/logs/apache2/$dominio/awstats_$dominio.awlog
if   !   ssh   -o   ConnectTimeout=2   root@10.0.3.100   perl /var/www/awstats/$dominio/awstats.pl -update -config=awstats.$dominio
then
ssh -o ConnectTimeout=2 root@10.0.3.101 perl /var/www/awstats/$dominio/awstats.pl -update -config=awstats.$dominio
fi
perl     /opt/script/logresolvemerge.pl     /data/logs/apache2/$dominio/*.log    >    /data/logs/apache2/$dominio/awstats_$dominio.awlog
```

Tarea cron *"synclogs.sh dominio.com"* programada para ejecutarse cada hora. Esta tarea programada se añade con el registro del dominio "dominio.com":

```
root@nfs1:~# crontab -l
30 2 * * * /opt/script/backup-data.sh
@hourly /opt/script/synclogs.sh dominio.com
```

A14. Gestión servicio FTP

Comandos usados en la gestión del servicio PureFTPd:

```
# Crear usuario
(-d --> directorio raiz usuarioftp // -N 100 --> cuota de 100 MB // -m --> reload del servicio)
(echo $passwordftp; echo $passwordftp) | pure-pw useradd $usuarioftp -d /data/$dominio -u ftpuser -N 100 -m

# Eliminar usuario
pure-pw userdel $usuarioftp -m

# Modificar password
(echo $passwordftp; echo $passwordftp) | pure-pw passwd $usuarioftp -m
```

A15. Redirección web

Comandos utilizados para el cambio de protocolo de HTTP a HTTPS y viceversa:

```
# Eliminar virtualhost del dominio con protocolo HTTP
rm -f /data/webconfig/sites-enabled/4$guion$dominio.conf

# Crear enlace simbólico en virtualhost con protocolo HTTP
ln   -s   /etc/apache2/sites-available/4$guion$dominio.conf /data/webconfig/sites-enabled/4$guion$dominio.conf

# Eliminar virtualhost del dominio con protocolo HTTPS
rm    -f    /data/webconfig/sites-enabled/5$guion$dominio-ssl.conf

# Crear enlace simbólico en virtualhost con protocolo HTTPS
ln     -s     /etc/apache2/sites-available/5$guion$dominio-ssl.conf     /data/webconfig/sites-enabled/5$guion$dominio-ssl.conf
```

A16. Redirección web

Fragmento del script *virtualhost-redirect.sh* que muestra como se hace el cambio de redirección permanente (301):

```
#!/bin/bash
...
if [ "$chksubdominio" == "" ]
        then
        dominioBase="$(cut -d'.' -f1 <<<"$dominio").$(cut -d'.' -f2 <<<"$dominio")"
        archivoConf=$sitesAvailable\4_$dominioBase.conf
        archivoConfSSL=$sitesAvailable\5_$dominioBase-ssl.conf
        else
        subdominio="$(cut -d'.' -f1 <<<"$dominio")"
        dominioBase="$(cut -d'.' -f2 <<<"$dominio").$(cut -d'.' -f3 <<<"$dominio")"
        archivoConf=$sitesAvailable\2_$dominioBase\_$subdominio.conf
        archivoConfSSL=$sitesAvailable\3_$dominioBase\_$subdominio-ssl.conf
fi

if [ "$accion" == '-a' ] && [ "$tipo" == '301' ]
        then
                sed -i "s/\t.*Redirect 302 \/.*/\t#Redirect 302 \/ dominio.tld/g" $archivoConf
                sed -i "s/\t.*Redirect 302 \/.*/\t#Redirect 302 \/ dominio.tld/g" $archivoConfSSL
                sed -i "s/\t.*Redirect 301 \/.*/\tRedirect 301 \/ ${destino}/g" $archivoConf
                sed -i "s/\t.*Redirect 301 \/.*/\tRedirect 301 \/ ${destino}/g" $archivoConfSSL
fi
...
```

A17. Panel de control UAB Hosting

Captura de la pantalla donde se muestra el formulario de registro:

Captura de la pantalla donde se muestra el menú principal con sus respectivas opciones: